Model Development Phase

Date	09 July 2024
Team ID	SWTID1719999219
Project Title	Crystal Clear Vision: Revolutionizing Cataract Prediction through Transfer Learning Mastery
Maximum Marks	5 Marks

Model Selection Report

In the model selection report for future deep learning and computer vision projects, various architectures, such as CNNs or RNNs, will be evaluated. Factors such as performance, complexity, and computational requirements will be considered to determine the most suitable model for the task at hand.

Model Selection Report:

Model	Description
Resnet	ResNet-50 is a deep convolutional neural network with 50 layers that uses residual connections to efficiently train very deep networks by mitigating the vanishing gradient problem. Its architecture includes convolutional layers, batch normalization, and ReLU activations, organized into residual blocks, enhancing feature extraction and overall performance. ResNet-50 achieves state-of-the-art results in image recognition tasks and is widely used for transfer learning, making it ideal for complex computer vision tasks like object detection and segmentation
Vgg16	Vgg16 is a deep convolutional neural network with 16 layers, including 13 convolutional and 3 fully connected layers. It uses small 3x3 convolutional filters and max-pooling layers to capture complex features while

	maintaining simplicity and uniform architecture. VGG16 is known for its strong performance in image classification tasks and is frequently used for transfer learning. Its straightforward design and robust feature extraction capabilities make it suitable for various image processing applications
EfficientNet B1	EfficientNet-B1 is part of the EfficientNet family, which is designed to achieve high accuracy with optimized efficiency. EfficientNet-B1 uses the compound scaling method to balance the depth, width, and resolution of the network, resulting in a model that performs well while requiring fewer computational resources. This version of EfficientNet has fewer parameters and FLOPs compared to many other deep learning models, making it suitable for applications where both accuracy and efficiency are important, such as mobile and edge devices. EfficientNet-B1 is particularly effective for image classification tasks and is known for its excellent performance on benchmarks with reduced computational cost.
Inception V3	InceptionV3 is a deep convolutional neural network architecture that builds upon the original Inception model (GoogLeNet). It employs a series of Inception modules, which consist of parallel convolutional layers with different filter sizes, allowing the network to capture multi-scale features efficiently. InceptionV3 introduces additional improvements such as factorized convolutions, batch normalization, and label smoothing, enhancing both performance and computational efficiency. This architecture achieves high accuracy on image classification tasks with a relatively lower number of parameters and FLOPs compared to other models. InceptionV3 is well-suited for applications that require robust feature extraction and efficient computation, making it a popular choice for various computer vision tasks.

Final accuracy -

Resnet50-

```
(None, None, None, 2048)
(None, 2048)
Epoch 1/50
10/10 [==============================] - ETA: 0s - loss: 0.4960 - accuracy: 0.7750/usr/local/lib/python3.10/dist-packages/keras/src/engine
  saving_api.save_model(
10/10 [==============================] - 51s 3s/step - loss: 0.4960 - accuracy: 0.7750 - val_loss: 0.4466 - val_accuracy: 0.8000
Epoch 2/50
10/10 [==============================] - 36s 2s/step - loss: 0.3761 - accuracy: 0.8219 - val_loss: 0.3689 - val_accuracy: 0.8250
Epoch 3/50
10/10 [==============================] - 37s 2s/step - loss: 0.3223 - accuracy: 0.8906 - val_loss: 0.3560 - val_accuracy: 0.8500
Epoch 4/50
10/10 [==============================] - 32s 2s/step - loss: 0.3046 - accuracy: 0.8781 - val_loss: 0.3775 - val_accuracy: 0.8500
Epoch 5/50
10/10 [==============================] - 37s 2s/step - loss: 0.2929 - accuracy: 0.8687 - val_loss: 0.3559 - val_accuracy: 0.8750
Epoch 6/50
10/10 [==============================] - 32s 1s/step - loss: 0.2997 - accuracy: 0.8719 - val_loss: 0.4078 - val_accuracy: 0.8500
Epoch 7/50
10/10 [==============================] - 34s 2s/step - loss: 0.2825 - accuracy: 0.8813 - val_loss: 0.3455 - val_accuracy: 0.8750
Epoch 8/50
10/10 [==============================] - 29s 1s/step - loss: 0.2568 - accuracy: 0.9031 - val_loss: 0.3465 - val_accuracy: 0.8750
```

Vgg16-

```
Epoch 5/50
20/20 [==============================] - ETA: 0s - loss: 0.2208 - accuracy: 0.8969
Epoch 5: val_loss did not improve from 0.35826
20/20 [==============================] - 211s 10s/step - loss: 0.2208 - accuracy: 0.8969 - val_loss: 0.4516 - val_accuracy: 0.8500
Epoch 6/50
20/20 [==============================] - ETA: 0s - loss: 0.2097 - accuracy: 0.9094
Epoch 6: val_loss did not improve from 0.35826
20/20 [==============================] - 209s 10s/step - loss: 0.2097 - accuracy: 0.9094 - val_loss: 0.3616 - val_accuracy: 0.9000
Epoch 7/50
20/20 [==============================] - ETA: 0s - loss: 0.2160 - accuracy: 0.9094
Epoch 7: val_loss did not improve from 0.35826
20/20 [==============================] - 208s 10s/step - loss: 0.2160 - accuracy: 0.9094 - val_loss: 0.5668 - val_accuracy: 0.7500
Epoch 8/50
20/20 [==============================] - ETA: 0s - loss: 0.2170 - accuracy: 0.9062
Epoch 8: val_loss did not improve from 0.35826
20/20 [==============================] - 214s 11s/step - loss: 0.2170 - accuracy: 0.9062 - val_loss: 0.4116 - val_accuracy: 0.8500
Model saved at: /content/repository/yiweichen04-retina_dataset-914b0f4/model/vgg16_model.h5
```

Efficient B1 –

```
Epoch 1/50
10/10 [==============================] - ETA: 0s - loss: 0.5833 - accuracy: 0.7031/usr/local/lib/python3.10/dist-packages/keras/src/eng
  saving_api.save_model(
10/10 [==============================] - 49s 2s/step - loss: 0.5833 - accuracy: 0.7031 - val_loss: 0.5492 - val_accuracy: 0.7500
Epoch 2/50
10/10 [==============================] - 29s 1s/step - loss: 0.4924 - accuracy: 0.7594 - val_loss: 0.4763 - val_accuracy: 0.7500
Epoch 3/50
10/10 [==============================] - 31s 1s/step - loss: 0.4256 - accuracy: 0.8094 - val_loss: 0.4104 - val_accuracy: 0.8000
Epoch 4/50
10/10 [==============================] - 28s 1s/step - loss: 0.3824 - accuracy: 0.8562 - val_loss: 0.4005 - val_accuracy: 0.8000
Epoch 5/50
10/10 [==============================] - 27s 1s/step - loss: 0.3291 - accuracy: 0.8594 - val_loss: 0.3827 - val_accuracy: 0.8000
Epoch 6/50
10/10 [==============================] - 28s 1s/step - loss: 0.3217 - accuracy: 0.8719 - val_loss: 0.3631 - val_accuracy: 0.8250
Epoch 7/50
10/10 [==============================] - 27s 930ms/step - loss: 0.3006 - accuracy: 0.8813 - val_loss: 0.3517 - val_accuracy: 0.8250
Epoch 8/50
10/10 [==============================] - 26s 1s/step - loss: 0.2921 - accuracy: 0.8906 - val_loss: 0.3606 - val_accuracy: 0.8250
Epoch 9/50
10/10 [==============================] - 26s 1s/step - loss: 0.2749 - accuracy: 0.8875 - val_loss: 0.3477 - val_accuracy: 0.8250
```

Inception V3-

```
Epoch 1/50
10/10 [==============================] - ETA: 0s - loss: 5.6772 - accuracy: 0.7219 /usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3
  saving_api.save_model(
10/10 [==============================] - 386s 36s/step - loss: 5.6772 - accuracy: 0.7219 - val_loss: 821.4874 - val_accuracy: 0.7500
Epoch 2/50
10/10 [==============================] - 411s 41s/step - loss: 1.6518 - accuracy: 0.8281 - val_loss: 152897.5312 - val_accuracy: 0.7500
Epoch 3/50
10/10 [==============================] - 420s 41s/step - loss: 0.4699 - accuracy: 0.8500 - val_loss: 19500656.0000 - val_accuracy: 0.7500
Epoch 4/50
10/10 [==============================] - 347s 34s/step - loss: 0.3396 - accuracy: 0.8656 - val_loss: 16894314.0000 - val_accuracy: 0.7500
Model saved at: /content/best_inception_model.h5
2/2 [==============================] - 13s 2s/step - loss: 878.6385 - accuracy: 0.7500
Test accuracy: 0.75
2/2 [==============================] - 14s 2s/step
```

Final Accuracy table-

Model	Train accuracy	Validation accuracy
Resnet 50	90.31%	87.50%
Vgg16	90.62%	85%
Efficient B1	88.75%	82.5%
Inception v3	86.56%	75%

Model selected for web deployment of the project is Resnet 50.

As, Resnet50 and Vgg16 have a comparable train accuracy of 90%.But Resnet50 Validation accuracy is 87.5% and Vgg16 Validation accuracy is 85%